**CSC1035 – Semester 2 – Coursework 2 – C3021739**

**Outline of contributions/responsibilities**

Originally my responsibility in this group project was to create the questions class, the basic function of this class would allow for the creation of a set of questions which could then be saved to a file. This saved data would have the ability to be loaded again in the future. Further methods were needed for the user to get specific details of the questions, e.g. getting all the question IDs of a specific type to allow for the quiz to later be able to present this data to the user. I was also responsible for testing my own class to make sure correct data is processed correctly and incorrect data causes exceptions to be thrown.

However, I had to later pick up the quiz class since I was the quickest to finish my part of the project and therefore the most available. The quiz class takes the questions file generated by the questions class and shows them to the user, allowing them to answer the questions one by one and letting them exit the quiz at any point and resume later if they wished. At the end of the quiz, the user is shown the correct answers and the time it took them to answer the questions. The answers they provide are stored in a user log table alongside both the incorrect and correct answer’s question IDs for revision purposes.

**Reflection**

**What happened? What were your thoughts and feelings at the time?**

I had to complete my section in the first week because other group members needed it finished to work on their parts of the project. This worked out well because it gave me a lot more time in the second week to thoroughly test the questions class. Then we ran into an issue with a group member who wasn’t contacting us for a while, meaning that I had to pick up some of the work they were meant to do, we decided that I would create the quiz class and have someone else test it. This may seem more uneven workload-wise than it was since a large portion of the code in questions and quiz are getter methods which didn’t take a lot of time to create.

**Analysis and evaluation**

Overall, our project wasn’t organised very well due to failures in planning it out well on the first day. The responsibilities we assigned ended up being uneven in workload, and we were also sometimes unclear on the specific methods/tasks each person needed to complete. Bad organisation aside, I think our best strength was how at least four of us always showed up to the practical sessions each time allowing us to communicate on how things were going, seeing if people needed support, and setting out what work needed to be done the following week. In terms of the programming itself I think it went well, I was very good at committing often (after every bug fix or method creation) meaning that my part of the project had good version control, this is important in case of unexpected errors. However, I wish I were able to do more testing and error checking but unfortunately I ran out of time.

**Action plan for future group projects**

Poor organisation was our biggest problem in this group project and going forward, to combat this issue, I have some improvements I could make next time I work with others on a project. To begin, it’s very important that in the first session we have a concrete idea of who has which responsibility, this means that we should create a detailed diagram and list out the specific tasks/methods that would need to be completed in each class. This prevents the problems of unclear responsibilities and the uneven workload (since we’d more easily be able to tell how much work each task would be). Furthermore, I could be more proactive with checking in, seeing how the others are doing and asking if they need extra support or if they’re stuck on anything. This would help quicken the workflow and make other group members more willing to reach out for help if they needed it, this would also allow me to do more testing and error checking for my code.

**Log tables:**

|  |  |  |
| --- | --- | --- |
| File/Method Created/Modified | Date | Description of work |
| **Questions/TestQuestions class** | | |
|  |  |  |
| Created .gitignore file | 29/02/24 | Created .gitignore file and ignored .idea, .iml and out folders. |
| Added loadDatabase and saveDatabase methods | 07/03/24 | Methods for saving data to/loading data from csv files. |
| Added getQuestionIDs method | 07/03/24 | Method for returning a list of all question IDs stored. |
| Added addQuestion method | 07/03/24 | Method for adding question to the class through using the questions array list. |
| Added modifyType, modifyAnswer and modifyQuestion methods | 08/03/24 | Allows for question types/answers/questions to be modified. |
| Added removeQuestion method | 08/03/24 | Allows for questions to be removed. |
| Added checkType and checkQuestion | 08/03/24 | Checks if the type matches the answer.  Checks if the question exists. |
| Added Javadoc and comments | 08/03/24 | Added for readability, particularly for my group members. |
| Added getQid method | 08/03/24 | Returns the question ID corresponding to the question passed in. |
| Added getAllQuestions and getQIDsByType methods | 08/03/24 | Returns all questions stored, the other returns the QIDs corresponding to a particular type. |
| Override equals, hashCode and toString methods | 08/03/24 | This prevents unexpected errors from occurring and to return a string representation of an object. |
| Added tests for all methods in the Questions class (in TestQuestions) | 14/03/24 | Testing that inputting correct data works as expected and that incorrect data throws an exception. |
| Added comments/Javadoc to TestQuestions | 15/03/24 | Explains how the code works and what each test proves. |
| **Quiz class** | | |
| Added takeQuiz and answerQuiz methods | 17/03/24 | Displays each question to the user, allows them to answer, and then checks if its correct. |
| Added generateUserLogs and changeFile method | 17/03/24 | Allows user logs to be created which log both their correct and incorrect answers for each user. |
| Added quizPass method | 20/03/24 | Returns true if the user has a percentage of correct answers above 40%. |